

科目名	システムプログラミング Systems Programming			担当教員	篠山 学		
学 年	4 年	学 期	通年	履修条件	選択	単位数	2
分 野	専門	授業形式	講義・演習	科目番号	12237026	単位区別	履修
学習目標	オペレーティングシステムの役割を理解し、それらの資源を利用する手段としてシステムコールを使用したプログラミングが行える。計算機内でのプロセスの状態遷移を把握でき、それを自由にコントロールできるようになる。						
進め方	各学習項目ごとに内容の解説を行い、関連する例題を説明した後、実際に実行結果を確認し理解させる。その後課題プログラムを作成し、レポートとして提出する。必要な関数の用法等はオンラインマニュアル等の参照によって自ら解決できるよう指導する。						
学習内容	学習項目 (時間数)			学習到達目標			
	1. システムプログラミングのための C 言語知識 (6)			低学年で学んだ C 言語の知識に加えて、構造体、ポインタ、リスト処理について復習し理解を深める。D2:1, 2			
	2. UNIX におけるマルチプログラミングとプロセスの状態遷移 (2)			どのようにマルチプログラミングが実現されるかを理解する。プロセスの生成、プログラムの実行、他のプロセスとの同期を行うシステムコールを使った 20 行程度のプログラムはマニュアルを参照しながら作成できる。D2:2, 3)			
	3. fork, wait, exec, exit (基本概念) (2)						
	4. fork, wait, exec, exit (プログラミング) (6)						
	----- [前期中間試験] (2)						
	5. 試験問題の解答と授業評価アンケート ファイルシステム (基本概念) (2)			UNIX ファイルシステムのディスクの領域管理、ファイル管理の仕組みが理解できる。			
	6. ファイルシステム (プログラミング) (6)			ファイルの管理情報を参照、変更を行うプログラムを作成できる。D2:2, 3			
	7. 簡易シェルの作成 (4)			UNIX のシェルの位置付けを理解し、簡易なシェルを作成できる。D2:1-3			
	前期末試験						
8. 試験問題の解答と pipe 機能 (4)			標準入力とパイプラインが理解でき、プロセス間通信が行えるプログラムが作成できる。				
9. パイプ機能を持つシェルの作成 (6)			D2:1-3				
10. シグナルの原理 (基本概念) (4)			シグナルを使った、プロセス間の同期の原理を理解するとともに、プログラムが作成できる D2:2, E2:2				
11. シグナルの原理 (プログラミング) (6)							
----- [後期中間試験] (2)							
12. 試験問題の解答と共有メモリによる IPC (基本概念) (6)			共有メモリおよびメッセージを用いた複数プロセス間の通信の仕組みを理解し、プログラムが作成できる。D2:1, 2, E2:2				
13. メッセージによる IPC (基本概念) (6)							
後期末試験							
14. 試験問題の解答 (1)							
評価方法	定期試験 80%, レポート 20% の比率で評価する。						
履修要件	情報処理 II, ソフトウェア設計論						
関連科目	情報処理 II, 基礎情報工学, 計算機システム						
教 材	教科書: 羽山博 著 「Linux システムプログラミング」						
備 考	特になし						